Freeform Search

Database: US Pre-Grant Publication Full-Text Database
US Patents Full-Text Database
US OCR Full-Text Database
EPO Abstracts Database
JPO Abstracts Database
Derwent World Patents Index
IBM Technical Disclosure Bulletins

Term: L8 and (download$ with NIC)

Display: 10 **Documents in Display Format:** KWIC **Starting with Number** 1

Generate: ○ Hit List ◉ Hit Count ○ Side by Side ○ Image

Search Clear Interrupt

Search History

DATE: Friday, February 18, 2005 Printable Copy Create Case

Set Name side by side	Query	Hit Count	Set Name result set
	DB=USPT; PLUR=YES; OP=ADJ		
L10	L9 and (NIC with peripheral$)	2	L10
L9	L8 and (download$ with NIC)	11	L9
L8	709/$.ccls.	17245	L8
L7	L2 download$	0	L7
L6	L2 and media	2	L6
L5	L2 and device	1	L5
L4	L2 device	0	L4
L3	L2 and (NIC with peripheral$)	0	L3
L2	(5778180 or 5764896).pn.	2	L2
L1	(5778180 or5764896).pn.	0	L1

download w/ C

END OF SEARCH HISTORY

h e b b g e e e e c f ff e ch

Previous Doc Next Doc Go to Doc#
First Hit Fwd Refs

Generate Collection

L9: Entry 10 of 11 File: USPT Aug 8, 2000

DOCUMENT-IDENTIFIER: US 6101180 A
TITLE: High bandwidth broadcast system having localized multicast access to broadcast content

Detailed Description Text (99):
The controller unit 440 handles software downloads for itself and for all of the transponder units 445. Software downloads are preferably performed using FTP file downloads over the local ISP LAN the 240 through NIC 467, from a remote station over the modem interface 470, or through the RS-232 port 487. Before a file is downloaded, FTP server software in the controller unit 440 verifies that the download is, in fact, a new file. The files are preferably downloaded into a fixed directory structure.

Current US Cross Reference Classification (7):
709/203

Previous Doc Next Doc Go to Doc#

US006101180A

United States Patent [19]

Donahue et al.

[11] Patent Number: 6,101,180

[45] Date of Patent: Aug. 8, 2000

[54] **HIGH BANDWIDTH BROADCAST SYSTEM HAVING LOCALIZED MULTICAST ACCESS TO BROADCAST CONTENT**

[75] Inventors: **Paul W. Donahue; Jeffrey A. Dankworth; Larry W. Hinderks,** all of Reno, Nev.; **Laurence A. Fish; Ian A. Lerner,** both of San Diego, Calif.; **Thomas C. Ballister,** Reno, Nev.; **Roswell R. Roberts, III,** San Diego, Calif.

[73] Assignee: **Starguide Digital Networks, Inc.,** Reno, Nev.

[21] Appl. No.: **08/969,164**

[22] Filed: **Nov. 12, 1997**

Related U.S. Application Data

[60] Provisional application No. 60/029,427, Nov. 12, 1996, provisional application No. 60/039,672, Feb. 28, 1997, and provisional application No. 60/057,857, Sep. 2, 1997.

[51] **Int. Cl.**[7] **H04Q 11/04**

[52] **U.S. Cl.** **370/352**; 370/270; 370/337; 370/347; 370/389; 370/432; 370/486; 395/200.33; 395/200.8; 348/6

[58] **Field of Search** 370/260, 347, 370/348, 389, 395, 396, 397, 400, 401, 432, 449, 471, 352, 270, 486, 337, 342, 344, 429, 402; 395/200.33, 200.45, 200.47, 200.57, 200.59, 200.6, 200.8, 200.161, 200.62, 200.68, 200.79; 348/6, 7, 10, 12, 13; 379/90.01, 93.07

[56] **References Cited**

U.S. PATENT DOCUMENTS

4,018,993	4/1977	Edstrom	179/15 BY
4,437,907	7/1995	Picazo et al.	395/200
4,933,936	6/1990	Rasmussen et al.	370/85.9

(List continued on next page.)

FOREIGN PATENT DOCUMENTS

WO 97/48051 12/1997 WIPO G06F 13/00

OTHER PUBLICATIONS

Intel, "Intel ProShare Video Conferencing—White Paper," Jan. 10, 1997, pp. 1–5.

Videotex, "Multicast Video Service," Jan. 10, 1997,pp. 1–3

Sepmeier, "Internet Connectivity by Satellite," Feb. 6, 1997, pp. 1–2 (note 1996 copyright notice on p.2).

StarDust Technologies, Inc, "IP Multicast Initiative," Copyright 1995–96, IP Multicast Glossary pp. 1–7, IP Multicast BackGrounder, pp. 1–9, How IP Multicast Works, pp. 1–12.

Digex, "Digex to Supply Internet Connectivity to Southwestern Bell Internet Services," Oct. 3, 1996, one page.

Digex, "Orion Atlantic Launches International Internet Access Service Through Agreement With Digex," Sep. 17, 1996, one page.

Digex, "Amtrack Selects Digex to Track Onto the Internet," Aug. 23, 1996, one page.

Digex, "Digex Goes National . . . " Jul. 2, 1996, one page.

Digex, "Digex, Inc., and Winstar Communications, Inc. Form Partnership . . . ," Jun. 26, 1996, one page.

Digex, "LCI International/Digex Collaborate . . . ," Jun. 4, 1996, one page.

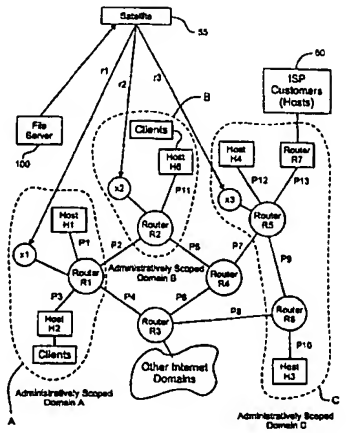
(List continued on next page.)

Primary Examiner—Douglas W. Olms
Assistant Examiner—Shick Hom
Attorney, Agent, or Firm—Robert C. Ryan; Nixon & Vanderhye P.C.

[57] **ABSTRACT**

A method of multicasting digital data to a user accessing an Internet connection is disclosed. The method includes placing digital data that is to be multicast in IP protocol to generate IP digital data. The IP digital data is transmitted from a transmission site to a remote Internet point of presence through a dedicated transmission channel substantially separate from Internet backbone. The dedicated transmission channel may be, for example, a satellite channel. At the remote Internet point of presence, the IP digital data is multicast for delivery to at least one receiving Internet user's apparatus connected to but distal from the remote Internet point of presence.

8 Claims, 33 Drawing Sheets



Previous Doc Next Doc Go to Doc#
First Hit Fwd Refs

Generate Collection

L9: Entry 11 of 11 File: USPT Oct 26, 1999

DOCUMENT-IDENTIFIER: US 5974547 A
TITLE: Technique for reliable network booting of an operating system to a client computer

Brief Summary Text (19):
While the boot process is occurring but prior to the availability of any client O/S-based network support, client hard disk emulation occurs through appropriate calls made to an interrupt (Interrupt 13 or simply "Int 13") handler. Through such calls, appropriate sectors in the client image file are initially downloaded, via a real-mode network adapter (NIC) driver and the Int 13 Handler to remotely install various components of the O/S into client PC. The actual client hard disk emulation process is provided through a real mode procedure that executes as part of Int 13 Handler. In essence, the real mode procedure determines, based on values of status flags, whether the client O/S is then capable of handling a network request for sector access of the client image file. If the client O/S has not then progressed to that point in its boot process, the real mode procedure processes that request, in real mode, through the Int 13 Handler.

Detailed Description Text (14):
As shown, client PC 10 comprises input interfaces (I/F) 310, processor 320, NIC 360, memory 330 and output interfaces 340, all conventionally interconnected by bus 350. Memory 330, which generally includes different modalities, includes illustratively random access memory (RAM) 332 for temporary data and instruction store, diskette drive(s) (not specifically shown) for exchanging information, as per user command, with floppy diskettes, and non-volatile mass store 335 that is implemented through hard disk drive(s) 334, typically magnetic in nature. Should client PC 10 be implemented by "diskless" computer, then all disk drives, including both floppy diskette drive(s) and hard disk drive(s) 334, would be omitted. Regardless of whether client PC 10 contained a hard disk drive or not, the client O/S, during its boot process, would be downloaded into RAM 332 and executed therefrom. As shown above in FIG. 2A, NIC 360 contains internal read-only memory 362, that stores network boot code 364. This code, as will be discussed shortly below, once downloaded into RAM 332 on power-up permits the NIC to establish a network connection to a remote server.

Detailed Description Text (19):
Once a user has powered-up client PC 10, as symbolized by block 420, the stored ROM BIOS in the client PC is loaded into RAM 332 (see FIG. 3) of the client PC from which that code is then executed by the PC. This operational mode is denoted by block 425 shown in FIGS. 4A and 4B. At this point, as symbolized by block 430, the client PC is not aware of its IP address. The client PC then reads the boot code from a ROM situated on the NIC (or alternatively on the motherboard of the client PC itself) into RAM 332 and then executes that code--this operational mode denoted by block 450. In response to this code, the client PC will broadcast, as symbolized by line 432, a BootP (or DHCP) request packet over the network to elicit a response from a BootP (or DHCP) server. Illustratively, server 50 contains BootP server 232. This packet contains the hardware address of the NIC. For exemplary purposes, I will assume that address is "00A024Baf9a5". BootP server 232, which is a conventional TCP server, permits a network device, such as the NIC, to obtain its

own IP address (i.e., here an IP address assigned to client PC 10), the name of a boot file to download, an IP address of a network server (here server 50) on which that boot file is located, and (where appropriate) an IP address of a default router. BootP server 232 does not download the boot file itself; that occurs, as will be shortly seen by TFTP server 402 executing within server 50. The IP address of the device can also be obtained through a DHCP request packet. DHCP is a newer protocol than BootP, and builds on and replaces BootP. Inasmuch as the BootP and DHCP protocols are conventional and well-known, I will not discuss them in any further detail. In that regard, for further information, the reader is referred to Chapter 19, "Booting Internet Hosts with BootP and TFTP" on pages 343-359 of P. Miller, TCP/IP Explained (.COPYRGT.1997, Digital Press)--hereinafter the "Miller" text; and Chapter 16, "BOOTP: Bootstrap Protocol" on pages 215-222 of W. R. Stevens, TCP/IP Illustrated, Volume 1--The Protocols (.COPYRGT.1994, Addison-Wesley Inc.). Both of these chapters are incorporated by reference herein. Since, for purposes of the present invention, either the BootP or DHCP protocols can be used with identical results, then, to simplify the ensuing discussion, I will omit any further reference to use of the DHCP protocol. The BootP server utilizes BOOTPTAB file 500. This file, illustratively shown in FIG. 5A, contains an entry for each of a number of remotely bootable devices that can connect to the network. Each such entry, such as entry 520 within entries 510, specifies for a single associated device: a hardware address (ha), i.e., a MAC, for that device; an associated boot file (bf) for that device; a home directory (hd) on that server which contains the boot file; and an IP address (ip) to assign to that device. For ease of access, the boot file and home directory reside on the same server as the BOOTPTAB file, here server 50. While the network may contain multiple BootP servers (of which, for simplicity, only one of which is shown in FIGS. 4A and 4B), each remotely bootable device, such as a given NIC, has only one unique corresponding entry in only one BOOTPTAB file. In this manner, a broadcast BootP request appearing on the network from a given device will engender only one response from a single server that has an entry, in its BOOTPTAB file, that contains a MAC matching that contained in the request.

Detailed Description Text (21):
In any event, after the client PC appropriately processes the BootP reply, the client PC will then know, as symbolized by block 440, its IP address. Next, as symbolized by line 442, the PC will issue, through the NIC, a TFTP request (typically a TFTP read command) to server 50, specifically TFTP server 402 thereon, to download the boot file identified in the BootP reply packet. If the TFTP server can locate and open this file based on the information provided in the TFTP request, then, as symbolized by line 444, TFTP server 402 will download the boot file to client PC 10. Once the boot file has been completely downloaded into RAM 332 (see FIG. 3) on client PC 10, this PC will acknowledge a successful download by issuing, as symbolized by line 446 shown in FIGS. 4A and 4B, a TFTP acknowledgement (ACK) packet, back to server 50. With the boot file (LANHD.IMG) residing, as symbolized by block 460, in the client PC and after the ACK packet is issued, the client PC will begin executing the boot file from RAM 332 to implement client hard disk emulation. At this point, client PC 10 begins operating, as symbolized by block 470, under control of the downloaded boot file (LANHD.IMG) and ceases operating under the ROM boot code previously downloaded from, e.g., the NIC.

Detailed Description Text (22):
The boot file, early in its execution, will cause the client PC to issue -a-TFPT request, as symbolized by line 462, back to server 50 to download an initialization file, specifically LANHD.INI file 550. This initialization file, as shown in FIG. 5B, also contains a series of entries. Here, each such entry, of which entry 560 is typical, contains a MAC, an IP address of a server that stores a client image file for the device having that MAC and a complete path to the client image file (here file 414 on server 410) on that server. A LANHD.INI file entry also contains an illustrative term, such as "3c90x" or "3c5x9", which merely describes a name of the NIC associated with that entry and is ignored, as a comment field, during

subsequent processing of this file by the client PC. A further parameter (db) in entry 560 defines a default boot option (illustratively set to A or C) which is not relevant here. Once the download, as symbolized by line 464, completes, client PC 10 will then generate and transmit, as symbolized by line 466, a TFTP ACK packet, over the network back to server 50. Furthermore, once this file has been downloaded, the client PC, under control of the boot file, LANHD.IMG, will process this file by first checking the contents of this file to determine whether an entry in the file contains a MAC that matches that of the NIC in the client PC. When an entry having a matching MAC is found, as illustratively occurs here, the boot file will then extract, from that entry, the full path to the client image file (here C: .backslash.LANHD.backslash.disk150) and the IP address (here 132.147.001.001) of a network server (here server 410) on which the client image file resides. Once client PC 10 obtains this information from the initialization file, the client PC, specifically executing the boot file (LANHD.IMG) which is then performing client hard disk emulation, issues, as symbolized by line 475, an RATFTP request to network server 410 to download a boot sector from client PC image file 414 residing thereon.

Detailed Description Text (26):
During the boot process and prior to the availability of any client O/S-based network support, client hard disk emulation occurs, as discussed above in conjunction with LANHD.IMG operation shown in FIGS. 4A and 4B, through appropriate calls made to Interrupt 13 (Int 13) handler 623. Through such calls, appropriate sectors in the client image file are downloaded, via real-mode NIC driver 625 and Int 13 Handler 623 to remotely install various components of O/S 690 into client PC. The actual client hard disk emulation process is provided through Real Mode Procedure 900 that executes as part of Int 13 Handler 623. In essence, procedure 900 determines, based on values of status flags, whether the client O/S is then capable of handling a network request for sector access of the client image file. If the client O/S has not then progressed to that point in its boot process, procedure 900 processes that request, in real mode, through Int 13. The remainder of this handler is conventional in nature.

Detailed Description Text (27):
In particular, as shown in start-up sequence 710, upon power-up of the client PC, this PC commences operation using real-mode processing, as symbolized by line 752. This mode of operation persists through downloading and initiation of the boot loader and client hard disk emulation code, i.e., boot file LANHD.IMG; and commencement of loading 32-bit client O/S 690 (e.g., Windows 95 O/S). Here, client hard disk emulation with sector-by-sector downloading is provided by block 620, specifically Int 13 Handler 623 and real-mode NIC driver 625.

Current US Cross Reference Classification (1):
709/217

Current US Cross Reference Classification (2):
709/220

Previous Doc Next Doc Go to Doc#

US005974547A

United States Patent [19]

Klimenko

[11] Patent Number: 5,974,547

[45] Date of Patent: Oct. 26, 1999

[54] **TECHNIQUE FOR RELIABLE NETWORK BOOTING OF AN OPERATING SYSTEM TO A CLIENT COMPUTER**

[75] Inventor: **Yevgeniy Klimenko**, Toronto, Canada

[73] Assignee: **3Com Corporation**, Santa Clara, Calif.

[21] Appl. No.: **09/045,577**

[22] Filed: **Mar. 20, 1998**

[51] **Int. Cl.**[6] **G06F 13/00**

[52] **U.S. Cl.** **713/2**; 713/100; 709/217; 709/220

[58] **Field of Search** 713/1, 2, 100; 709/220, 221, 222–228, 203, 217, 223, 224

[56] **References Cited**

U.S. PATENT DOCUMENTS

5,280,627	1/1994	Flaherty et al.	395/700
5,404,527	4/1995	Irwin et al.	395/700
5,574,915	11/1996	Lemon et al.	395/700
5,577,210	11/1996	Abdous et al.	709/200
5,644,714	7/1997	Kikinis	709/200
5,689,708	11/1997	Regnier et al.	709/302
5,842,011	9/1995	Basu	713/2
5,893,106	7/1997	Brobst et al.	707/102

OTHER PUBLICATIONS

P. Miller, *TCP/IP Explained* (© 1997, Digital Press), specifically Chapter 9, "Booting Internet Hosts with BootP and TFTP", pp. 343–359.

W. R. Stevens, *TCP/IP Illustrated, vol. 1—The Protocols* (© 1994, Addison–Wesley Inc.), specifically Chapter 16, "BOOTP: Bootstrap Protocol", pp. 215–222.

Primary Examiner—Joseph E. Palys

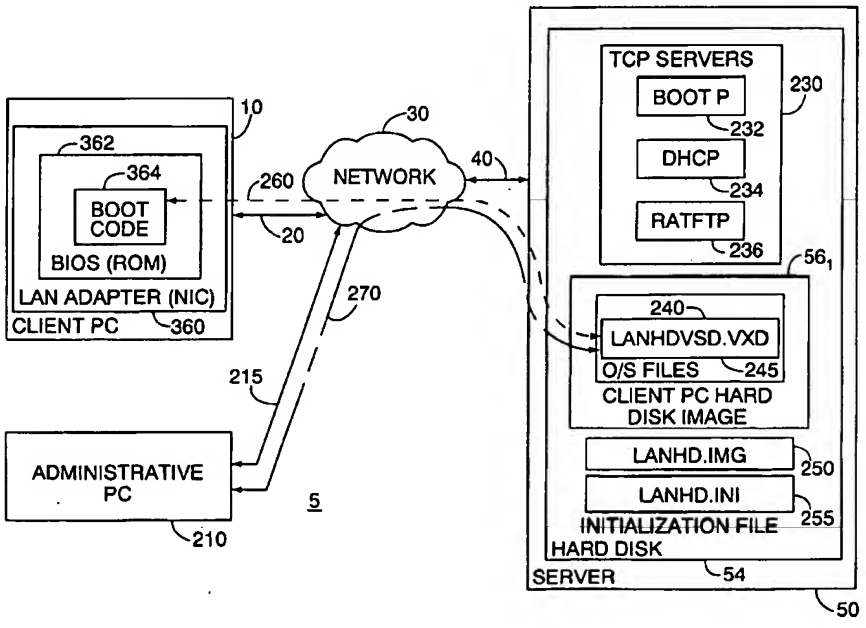
Assistant Examiner—Rijue Mai

Attorney, Agent, or Firm—Michaelson & Wallace; Peter L. Michaelson

[57] **ABSTRACT**

A technique, specifically apparatus and accompanying methods, for use in a client-server environment for booting an operating system (O/S), such as a 32-bit personal computer (PC) O/S, on a client computer through a networked connection to a server. Specifically, the server stores an image of a client hard disk including the client O/S and desired applications. During a boot process, a procedure, which is compliant with both an interrupt handler in the client and a network driver kernel in the client O/S, is installed in the client. Based on client O/S resources then available when, during the boot process, the client requests a local hard disk access to a particular sector, the procedure will re-direct that request, to the network file server, through a network driver kernel in the client O/S rather than through a client interrupt handler. Each such request is processed, to provide physical sector read or write access, through my inventive random access trivial file transfer protocol (RATFTP) server executing in the network server. Advantageously, the source of the sectors remains transparent to the client O/S, while it is being booted from a network connection, in lieu of a local hard disk drive. Hence, client hard disk emulation occurs seamlessly and continuously throughout the entire boot process even though, during this process, the client processing mode changes from real to protected and the client O/S resets and gains control of a client network adapter.

39 Claims, 14 Drawing Sheets



Name side by side	Query	Hit Count	Set Name result set
	DB=USPT; PLUR=YES; OP=ADJ		
L6	L2 and ((NIC or (network adj1 interface adj1 card)) adj3 (processor or CPU) adj3 includ$)	0	L6
L5	L2 and ((NIC or (network adj1 interface adj1 card)) adj3 (processor or CPU))	66	L5
L4	L2 and (NIC or (network adj1 interface adj1 card)).ti.	19	L4
L3	L2 and (NIC or (network adj1 interface adj1 card)).ab.	86	L3
L2	709/$.ccls.	17245	L2
L1	6070253.pn.	1	L1

END OF SEARCH HISTORY

L5/63
↑
NIC
+
Processor

h e b b cg b e e ch

Previous Doc Next Doc Go to Doc#
First Hit Fwd Refs

Generate Collection

L5: Entry 63 of 66 File: USPT Jun 9, 1998

DOCUMENT-IDENTIFIER: US 5764896 A
TITLE: Method and system for reducing transfer latency when transferring data from a network to a computer system

Brief Summary Text (9):
Data is typically transferred across network segments in the form of packets or frames. A NIC usually includes a buffer or the like for temporarily storing data transferred between a computer system and a network. A key parameter for determining the performance of data transmission is data transfer latency from the network to the computer system. Latency is a measure of the amount of time or delay to transfer a packet or packet portion to the main memory of the computer, and may further include the additional time to inform the host processor of the transfer, if necessary. In many Ethernet and token ring schemes prior to the present invention, an entire block of data was written into the buffer for temporary storage before being transferred to the main memory. Each block formed a portion of a packet or the entire packet and is typically approximately one kilobyte in length. Transfer of each block from the NIC to the main memory depended upon whether the NIC was capable of performing direct memory access (DMA) data transfers. If so, after a block was written to the buffer of the NIC, the NIC gained control of the expansion bus of the computer and performed a DMA transfer of the block into the computer memory. For memory-mapped configurations, the NIC informed the host processor, usually by interrupt, and the host processor controlled the transfer of the data from the buffer to the computer memory.

Current US Original Classification (1):
709/250

Current US Cross Reference Classification (1):
709/233

Current US Cross Reference Classification (2):
709/235

Previous Doc Next Doc Go to Doc#

h e b b g e e e f e h f e g

US005764896A

United States Patent [19]

Johnson

[11] Patent Number: 5,764,896

[45] Date of Patent: Jun. 9, 1998

[54] **METHOD AND SYSTEM FOR REDUCING TRANSFER LATENCY WHEN TRANSFERRING DATA FROM A NETWORK TO A COMPUTER SYSTEM**

[75] Inventor: **Scott C. Johnson**, Williamson County, Tex.

[73] Assignee: **Compaq Computer Corporation**, Houston, Tex.

[21] Appl. No.: **671,583**

[22] Filed: **Jun. 28, 1996**

[51] **Int. Cl.**[6] **G06F 13/00**

[52] **U.S. Cl.** **395/200.8**; 395/200.63; 395/200.65; 395/280; 395/282; 395/872

[58] **Field of Search** 395/280, 282, 395/200.2, 872-877, 200.62, 200.63, 200.64, 200.65

[56] **References Cited**

U.S. PATENT DOCUMENTS

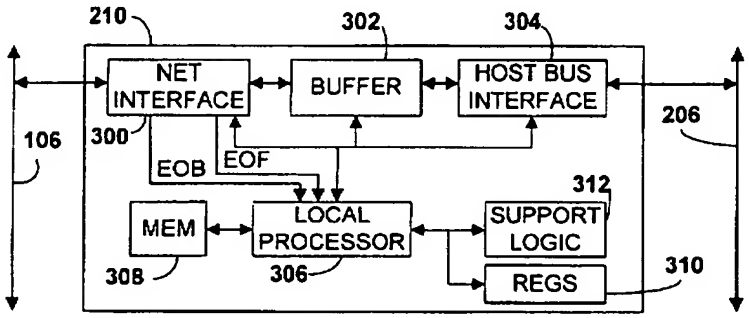
5,297,139	3/1994	Okura et al.	370/428
5,307,459	4/1994	Petersen et al.	395/200.8
5,434,872	7/1995	Petersen et al.	371/57.1
5,606,559	2/1997	Badger et al.	370/395

Primary Examiner—Glenn A. Auve
Assistant Examiner—Ario Etienne
Attorney, Agent, or Firm—Stanford & Bennett

[57] **ABSTRACT**

A computer system for communicating with a network including a host processor, memory, an interface bus and a network interface device for reducing data transfer latency between the computer system and the network. The network interface device includes a buffer for temporarily storing data, a media interface device for transferring data between the buffer and the network, a bus interface for transferring data between the computer system's memory and the buffer, and a local processor for writing a unique value at a predetermined location within the buffer, for periodically comparing the data value at the predetermined location with the unique value and for initiating data transfer from the buffer to the computer's memory when the data value does not match the unique value. The network interface device is preferably a network interface card (NIC) for plugging into a slot of the interface or expansion bus of the computer system. The local processor writes the unique value at the location and then periodically compares the data at that location with the unique value. When the data value is different from the unique value, the local processor has detected new data in that block of memory. The local processor responsingly initiates data transfer of the new data from that block to the memory of the computer system.

20 Claims, 3 Drawing Sheets



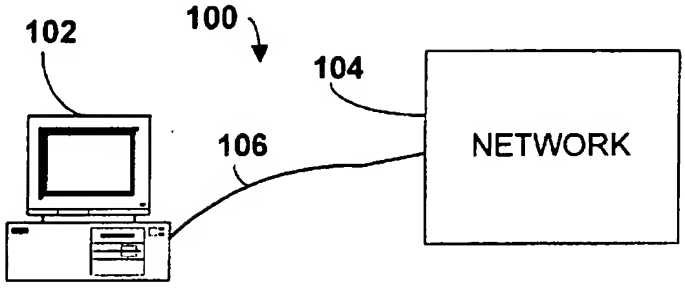


FIG. 1

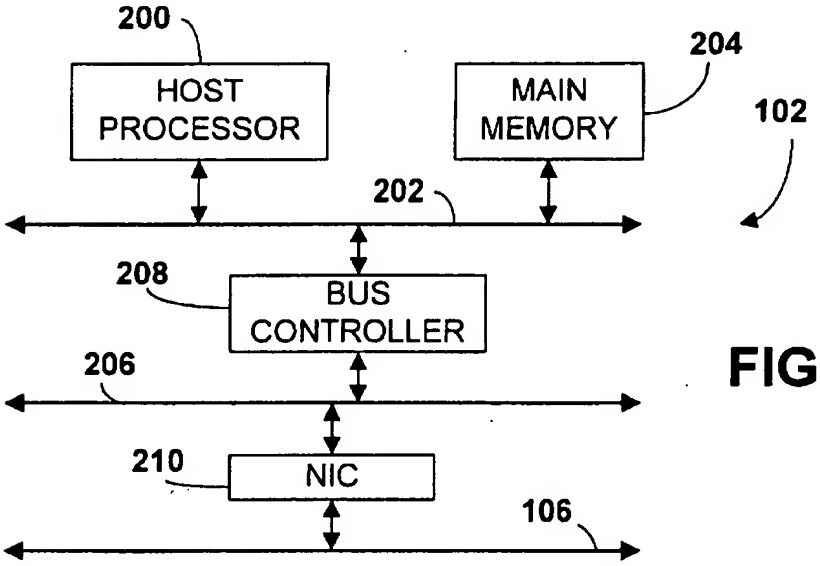


FIG. 2

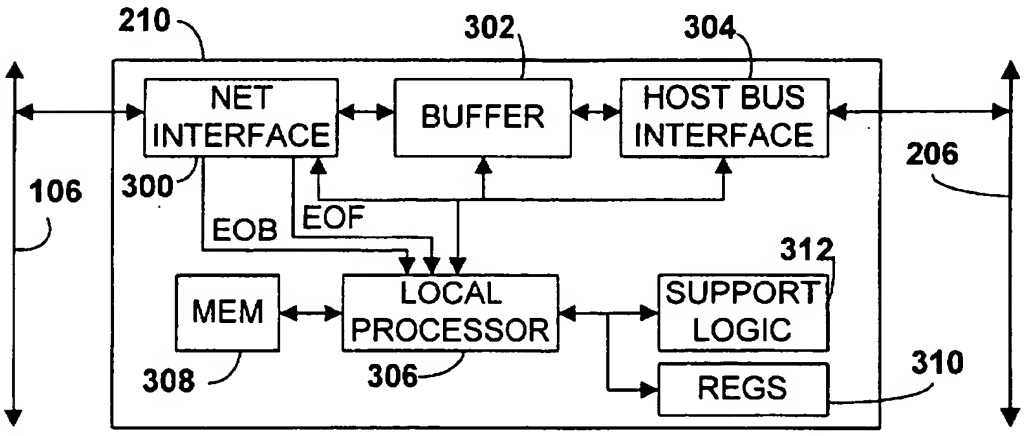


FIG. 3